

# User Documentation

## Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Task Manager</b>	<b>1</b>
<b>Task Board</b>	<b>2</b>
<b>Price Checker</b>	<b>3</b>
<b>Login Page</b>	<b>3</b>
<b>Checkout</b>	<b>4</b>
<b>Item Locator</b>	<b>4</b>

## Task Manager

After the Task Manager application is opened, the user is brought to the index page for the Task Board application. Here, the user has multiple options: (1) the option to create a task and (2) options for the editing of a given, already created task. The option to create a task is represented by the “Create Task” button at the top of the page. All tasks are shown in a table with the most recent task on top. The following attributes about the task are shown: the task name, the task description, the task ID, the state, the employee ID, the time of the task’s creation, and the time of the task’s completion. At the end of each row in the video is an “Edit” button to allow the user to select the option to edit the task.

If the user chooses to create a task, they will be taken to a page with two text boxes. The upper text box should contain the task name. The lower text box should contain the task description. After the two text boxes are filled in with the proper information, the user should click on the “Create Task” button. If the user opts to not create a task, the user may click on the “Back to Task List” button.

If the user opted to create a task and pressed the “Create Task” button, they will be redirected to the index page with a highlighted message at the top of the screen that the task has been created. The message may be removed from the page by pressing the “x” within the highlighted portion. The created task will appear in the table of tasks.

If the user selected a task to be edited, then they will be taken to a page showing the selected task and text boxes and a drop down menu for which attributes of the task can be changed. The following attributes may be altered: the task name, the task description, the state, the employee ID, the time of the task's creation, and the time of the task's completion. It should be noted that the state is changed by a selection in the drop down menu, but all other changes may be submitted by entering text in the text boxes under the attribute which is to be changed. Of additional note should be that the time of the task's creation and the time of the task's completion should be of a particular time and date format. After the relevant text boxes are filled out and the state is selected (if necessary), the user should click on the "Update Task" button. If the user does not want to modify the task and instead would like to delete the task, the user does not have to fill in any of the text boxes and should click on the "Delete Task" button. If the user opts to not alter a task at all, the user may click on the "Back to Task List" button.

If the user opted to update the task from the index page, they will be redirected to a page which will show in a table the task prior to the modification and the task after the modification. There is also a "Back to Task List" button that the user may click to return to the index page

If the user opted to delete a task and pressed the "Delete Task" button, they will be redirected to the index page with a highlighted message at the top of the screen that the task has been deleted. The message may be removed from the page by pressing the "x" within the highlighted portion. The deleted task will not appear in the table of tasks.

## **Task Board**

After the Task Board application is opened, the user is brought to the index page for the Task Board application. Here, the user has two options: (1) claim tasks or (2) view the user's own tasks. The claim tasks option is represented by the "Claim Task" button, the option to view tasks is represented by the "View Your Tasks" button. Currently, because login functionality has not yet been integrated into this application, an employee ID must be entered in order to view tasks.

If the user opted to claim tasks, they will be redirected to a page with a table of incomplete tasks that can be claimed. Each row represents a task. At the end of each row is an option to claim a task represented by "Claim" buttons. Currently, because login functionality has not yet been integrated into this application, an employee ID must be entered in order to claim tasks. If the user opts to not claim any task, the user may click on the "Back" button to return to the index page.

If the user claimed a task, they will be redirected to a page that shows the task that was claimed and that states the task was claimed. There is also a “Back” button that the user may click to return to the task claiming page to potentially claim more tasks.

If a user opted to view their tasks from the index page, the user will see a table of their tasks if they have claimed any tasks. At the end of each row, there is a “Complete” button to represent an option to mark a claimed task as complete. If the user opts to not complete any task, the user may click on the “Back” button to return to the index page.

If the user completed a task, they will be redirected to a page that shows the task that was completed and that states the task was completed. There is also a “Back” button that the user may click to return to the task viewing page to potentially complete more tasks.

## Price Checker

Running `pricecheckertest.py` will open a window with a demonstration of how the price checking functionality is implemented. The user is shown an input text field and a button labeled scan. In this demo, the input field takes in an RFID which identifies an item, and the Scan button begins the search for an item with the input RFID. In the final version of the price checker, the RFID will be retrieved by a scanner and the scanning process will start automatically afterwards.

In this demo, we have provided 15 test RFIDs for the user to input. Use the following ids as input (ids separated by semicolons).

00000; 11111; 22222; 33333; 55555; 99999; aaaaa; aaaa2; bbbb3; rfid0;  
12345; asdfg; ii2jc; cc1?1; lo6dk

Inputting any of these ids will result in a message detailing the price of the item to the user. Entering an invalid id will display the message “Error detecting item. Please scan again”. Though it is very unlikely an invalid id will be scanned at the store, this measure has been included in case of damaged RFID tags which no longer are able to provide a readable id.

## Login Page

The login function is implemented in the Employee and Manager classes. It accepts a username and password. Using the username, it calls a helper function that fetches the password from the

database. The entered password is checked against this and a boolean value is returned to authenticate employee/manager login.

It also has features to create a new employee account and a forgot password button.

## Checkout

Once the Checkout application is selected you are shown the current cart, which is a demonstration of putting down the first bag. That bag would then be removed and the second would be placed down. For demo purposes the “Add Items” button was created in order to simulate this function. The button will then add a second bag containing 2 items to the list which will then be displayed in the table. When you are done you can choose to “Finish” bringing you to a screen that will allow the user to choose which payment method, Cash or Credit, they would like. It would then redirect to the proper external system to complete the transaction.

## Item Locator

Item Locator helps customers find an item that they want. The function takes in input from the customers and returns a map of where that item is located with its aisle and section number. This is more convenient than customers looking around the store puzzled instead they would just search an item and in a few seconds get the location of that item. In *getLocation.py* customers enter an item then it checks if that item is in the store and that it is in stock. If both conditions are met then it prints out the aisle and section number of that item. However, if one of the conditions is not met then it prints out “Item out of stock”. Furthermore, *itemFinder.html* is an interface that allows customers to search items they want and it gives them the map of that item.